



CSW

PTO/SB/21 (08-03)  
Approved for use through 08/30/2003. OMB 0651-0031  
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

<b>TRANSMITTAL FORM</b>  (to be used for all correspondence after initial filing)	Application Number	10/761,697
	Filing Date	1/20/04
	First Named Inventor	Yuji Suga
	Art Unit	2131
	Examiner Name	
	Attorney Docket Number	CFA00044US
Total Number of Pages in This Submission	33	

ENCLOSURES (Check all that apply)		
<input type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment/Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input checked="" type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance communication to Technology Center (TC) <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input type="checkbox"/> Other Enclosure(s) (please identify below):
<div>Remarks</div>		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Canon U.S.A., Inc. IP Department Fidel Nwamu
Signature	
Date	5/5/04

CERTIFICATE OF TRANSMISSION/MAILING	
I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.	
Typed or printed name	Fidel Nwamu
Signature	
Date	5/3/04

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2003年 1月24日

出願番号  
Application Number: 特願2003-016718  
[ST. 10/C]: [JP2003-016718]

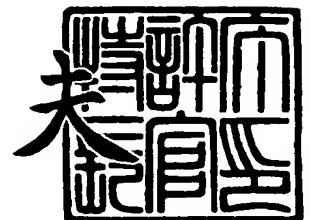
出願人  
Applicant(s): キヤノン株式会社

特許庁  
日本国特許庁  
特許庁  
特許庁

2004年 1月14日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



出証番号 出証特2003-3111145

【書類名】 特許願

【整理番号】 251199

【提出日】 平成15年 1月24日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 7/00

【発明の名称】 連鎖型署名作成方法

【請求項の数】 1

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社  
社内

【氏名】 須賀 祐治

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100076428

【弁理士】

【氏名又は名称】 大塚 康德

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100112508

【弁理士】

【氏名又は名称】 高柳 司郎

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100115071

【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

## 【選任した代理人】

【識別番号】 100116894

【弁理士】

【氏名又は名称】 木村 秀二

【電話番号】 03-5276-3241

## 【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0102485

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 連鎖型署名作成方法

【特許請求の範囲】

【請求項 1】 N 個の公開鍵とそのうちの 1 つの公開鍵に対応する秘密鍵によって作成可能で、N 個の公開鍵のそれぞれについて署名の検証が可能で、N 人のメンバーのうち誰が署名したかを秘匿することのできる連鎖型署名において、前記連鎖型署名データの作成者以外のユーザが署名していないことを検証できるようにするための否認データを生成する工程を備えることを特徴とする連鎖型署名作成方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は入力デジタルデータに対する連鎖型署名データを生成するために用いて好適なものである。

【従来の技術】

文書や画像データがインターネットなどの広域ネットワーク網を通して流通する場合、デジタルデータは改変が容易であるため、第三者によってデータが改ざんされる危険性がある。そこで、送信されてきたデータが改ざんされたかどうかを受信者が検出するために、改ざん防止用の付加データを検証する方式としてデジタル署名という技術が提案されている。デジタル署名技術は、データ改ざんだけではなく、インターネット上でなりすまし、否認などを防止する効果も持ち合わせている。

【0 0 0 2】

[デジタル署名]

デジタル署名データ生成にはハッシュ関数と公開鍵暗号とが用いられる。これは、秘密鍵を  $K_s$ 、公開鍵を  $K_p$  とすれば、送信者は、入力データ  $M$  にハッシュ処理を施して固定長データ  $H(M)$  を算出した後、秘密鍵  $K_s$  で上記固定長データ  $H(M)$  を変換してデジタル署名データ  $S$  を作成し、その後、デジタル署名データ  $S$  と入力データ  $M$  とを受信者に送信する。

**【 0 0 0 3 】**

受信者は、上記デジタル署名データ S を上記公開鍵  $K_p$  で変換（復号）したデータと、入力データ M にハッシュ処理を施したデータとが一致するかどうか検証する。そして、上記検証の結果が一致していなければ、データ M の改ざんが行われたことを検出できる。

**【 0 0 0 4 】**

デジタル署名には RSA、DSA など公開鍵暗号方式が用いられおり、署名の安全性は、秘密鍵の所有者以外のエンティティは、署名を偽造、もしくは秘密鍵を解読することが離散対数問題に帰着され、計算量的に困難であることに基づいている。

**【 0 0 0 5 】****[ハッシュ関数]**

次に、ハッシュ関数について説明する。ハッシュ関数はデジタル署名データの生成を高速化するため等に用いられる。ハッシュ関数は任意の長さのデータ M に処理を行い、一定の長さの出力データを生成する機能を持っている。ここで、出力  $H(M)$  を平文データ M のハッシュデータと呼ぶ。

**【 0 0 0 6 】**

特に、一方向性ハッシュ関数は、データ M を与えた時、 $H(M') = H(M)$  となる平文データ  $M'$  の算出が計算量的に困難であるという性質を持っている。上記一方向性ハッシュ関数としては MD2、MD5、SHA-1 などの標準的なアルゴリズムが存在しており、これらのアルゴリズムは公開されている。

**【 0 0 0 7 】****[公開鍵暗号]**

次に、公開鍵暗号について説明する。公開鍵暗号は 2 つの異なる鍵を利用し、片方の鍵で暗号処理したデータは、もう片方の鍵でしか復号処理できないという性質を持っている。上記 2 つの鍵のうち、一方の鍵は公開鍵と呼ばれ、広く公開するようにしている。また、もう片方の鍵は秘密鍵と呼ばれ、本人のみが持つ鍵である。

**【 0 0 0 8 】**

公開鍵暗号方式を用いたデジタル署名においては、次のように署名者を匿名にする技術が開発されている。ここでは、グループ署名とリング署名（連鎖型署名）について説明する。

#### 【0 0 0 9】

##### [グループ署名]

グループ署名とは、署名がグループのメンバーにより作成されたかどうかを誰でも検証できるが、誰が署名したかはわからない署名方式であり、1991年にChaumによって提案された方式である。メンバーのほかに特権を持つ管理者が存在し、問題が起こったときは、特別な方法で署名者を特定できる仕組みを備えている。

#### 【0 0 1 0】

グループ署名は、a) グループ公開鍵がグループに所属するメンバーの公開鍵のリストとなっている公開鍵登録型と、b) グループメンバーに対しメンバーである証明書を発行する証明書発行型の2つに大別することができる。

#### 【0 0 1 1】

a) の手法ではグループ公開鍵と署名のサイズがメンバー数に依存した大きさとなり非効率である。しかしグループからメンバーを削除するのは容易に行うことができる。

#### 【0 0 1 2】

b) の手法はグループ公開鍵と署名のサイズはメンバー数に依存しないが、メンバーの削除を行うには一度発行した証明書を無効にする必要があるという特徴を持つ。

#### 【0 0 1 3】

グループ署名は、利用者のプライバシーを保護する目的のアプリケーションである、電子決済プロトコルや、電子オークションプロトコルに応用されている。

##### [リング署名]

グループ署名方式では自分の身元を明かすことなくグループへの所属を証明することができるが、メンバーとは別に特権を持つ管理者を必要としていた。一方、リング署名は、このような管理者は必要なく、署名作成のためのメンバーとの

事前やり取りなどは全く必要のない方式で、2001年にShamirらによって提案された。

#### 【0014】

[Shamirらによるリング署名]

$\{0, 1\}^l$  の入出力を持つ落とし戸付き一方向性置換関数を  $g_0, \dots, g_{n-1}$  とする。 $H()$  は通常のハッシュ関数、 $E_K()$ 、 $D_K()$  を共有鍵  $K$  による共通鍵暗号の暗号化関数、復号化関数とする。署名作成者はある  $i$  における  $g_i$  の逆関数を秘密に保持しているとする。ここで  $x \oplus y$  は排他的論理和演算のことである。

#### 【0015】

[Shamirリング署名：署名作成]      文書  $M$  に対する署名の作成手順

1)  $K := H(M)$  とおく。

#### 【0016】

2)  $Z_0$  を  $\{0, 1\}^l$  から任意に選択する。

#### 【0017】

3)  $j = 0, \dots, i-1$  (昇順) について、以下を繰り返す。 $r_j$  を  $\{0, 1\}^l$  から任意に選択し、 $y_j := g_j(r_j)$ 、 $z'_j := z_j \oplus y_j$ 、 $z_{j+1} := E_K(z'_j)$  とおく。

#### 【0018】

4)  $z'_{n+1} := D_K(Z_0)$

5)  $j = n-1, \dots, i+1$  (降順) について以下を繰り返す。 $r_j$  を  $\{0, 1\}^l$  から任意に選択し、 $y_j := g_j(r_j)$ 、 $z_j := z'_{j+1} \oplus y_j$ 、 $z_{j-1} := D_K(z'_j)$  とおく。

#### 【0019】

6)  $g_i$  の逆関数を知る署名者が、以下を計算する。 $y_i := z_i \oplus z'_{i+1}$ 、 $r_i := g_i^{-1}(y_i)$

7) 署名  $(z_0, r_0, r_1, \dots, r_{n-1})$  を出力する。

#### 【0020】

[Shamirリング署名：署名検証]      文書  $M$  に対する署名  $(z_0, r$

$r_0, r_1, \dots, r_{(n-1)}$  の検証手順

1)  $K := H(M)$  とおく。

【0021】

2)  $j = 0, \dots, n-1$  (昇順) について、以下を繰り返す。 $y_j := g_j(r_j)$ 、 $z'_j := z_j \text{ xor } y_j$ 、 $z_{(j+1)} := E_K(z'_j)$  とおく。

【0022】

3)  $z_n = z_0$  かどうかを検証する。

【0023】

以上の手順では、様々な既存の署名方式に適用できる点が優れているが、a) 落とし戸付き一方向性置換関数、b) 共有鍵暗号、復号関数の2つが安全に提供されねばならない。

【0024】

[大久保らによるリング署名]

上記の問題を解決するために、a)、b) の関数を必要としない署名方法が提案されている。ただし、Schnorr 署名と呼ばれる既存の署名方式にのみ適用できる点で汎用性が低い。

【0025】

[Schnorr 署名]

従来技術としての Schnorr 署名を説明する (例えば、非特許文献 1 を参照)。

【0026】

$p, q$  を素数とし、 $p-1$  は  $q$  を割り切るとする。 $g$  を  $Z_p^*$  (位数  $p$  の巡回群  $Z_p$  から 0 を省いた乗法群) から任意に選択した、位数  $q$  の元 (生成元) とする。 $Z_p^*$  から選択した  $x$  を秘密鍵とし、それに対する公開鍵  $y$  を  $y := g^x \text{ mod } p$  とおく。 $H()$  をハッシュ関数とする。

【0027】

[Schnorr 署名作成] 文書  $M$  に対する署名の作成手順

1)  $\alpha$  を  $Z_q$  から任意に選択し、 $T := g^\alpha \text{ mod } p$  とおく。

## 【0028】

- 2)  $c := H(M || T)$  とおく。ただし  $||$  はデータの連結を意味する。

## 【0029】

- 3)  $s := \alpha - x c \bmod q$  とおき、 $(s, c)$  を署名データとする。

## 【0030】

[Schnorr 署名検証] 文書  $M$  に対する署名  $(s, c)$  の検証手順

$T := g^{s y c} \bmod p$  とおき、 $c = H(M || T)$  かどうか検証する。

## 【0031】

大久保らによるリング署名は、Schnorr 署名を逐次的に結合したものと捉えることができる。

## 【0032】

以下、従来技術としての Schnorr 署名によるリング署名を説明する（例えば、非特許文献 2 を参照）。

## 【0033】

記号の表記方法は Schnorr 署名での記号に準じる。署名者は、 $n$  個の公開鍵  $(g_i, p_i, q_i)$  に対する  $y_i$  を得ておく。そのうち、 $y_i$  に対する秘密鍵  $x_i$  を知っているものとする。 $H_i()$  はハッシュ関数とする。添字は  $\bmod n$  と取るものとする。例えば  $x_{(n+1)}$  は  $x_0$  とみなす。

## 【0034】

[Schnorr リング署名作成] 文書  $M$  に対する署名の作成手順

- 1)  $\alpha$  を  $Z_{(q_i)}$  から任意に選択し、 $T_i := g_i^\alpha \bmod p_i$  とおく。

## 【0035】

- 2)  $c_{(i+1)} := H(M || T_i)$  とおく。

## 【0036】

- 3)  $j = i + 1, \dots, i - 1$  (昇順) について、以下を繰り返す。 $s_j$  を  $Z_{(q_j)}$  から任意に選択し、 $T_j := g_j^{s_j y_j c_j} \bmod p_j$

$d_{p\_j}, c_{(j+1)} := H(M || T_{j})$  とおく。

【0037】

4)  $s_i := \alpha - x_i \cdot c_i \bmod q_i$  とおき、 $(c_0, s_0, s_1, \dots, s_{(n-1)})$  を署名データとする。

【0038】

[Schnorr リング署名検証] 文書Mに対する署名  $(c_0, s_0, s_1, \dots, s_{(n-1)})$  の検証手順

1)  $j = 0, \dots, n-1$  (昇順) について、以下を繰り返す。 $T_j := g_{s_j} y_{c_j} \bmod p_j, c_{(j+1)} := H(M || T_j)$  とおく。

【0039】

2)  $c_n = c_0$  かどうかを検証する。

【非特許文献1】

C.P.Schnorr, "Efficient Signature Generation by Smart Cards", Journal of Cryptology, Vol.4, No.3, pp.161-174 (1991)

【非特許文献2】

大久保, 安部, 鈴木, 辻井, "証明長が短い 1-out-of-n 証明", 4C-4, pp.189-193, 2002年 暗号と情報セキュリティシンポジウム(SCIS200)

【0040】

【発明が解決しようとする課題】

Shamirらによるリング署名、大久保らによる Schnorr リング署名は、管理者不在であることにより、自由に第3者の公開鍵を取得して、擬似的に署名を施すことで、匿名性を確保している。しかし、第3者の公開鍵を取得するだけでリングの中に擬似的な署名を含めることができるため、勝手に公開鍵を流用される可能性がある。この場合、勝手に流用された公開鍵に対する秘密鍵を保持するユーザは、自らが署名していないことを示す（否認する）ことができない問題が生じる。

【0041】

リング署名の具体的な適用例としては、報道機関に対する内部告発が挙げられ

る。告発者は自分の身元を明かすことなく、文書の信憑性を確保することができ  
る意味で有用である。しかし、告発者以外のリング署名内に含まれた者は、実際  
に告発していないにも関わらず疑いをかけられてしまう可能性がある。この場合  
、「自らが署名した文書ではない」ことを第3者に証明する手立てはない。

#### 【0042】

本発明は以上の問題に鑑みてなされたものであり、リング署名において、勝手に  
に流用された公開鍵に対する秘密鍵を保持するユーザ自身が署名を作成していな  
いことを示す否認データを作成することを目的とする。

#### 【0043】

また、この否認データは、リング署名の署名者は作成することができないよう  
にする必要がある。前述の例では、実際に告発したにも関わらず、「自らが署名  
した文書ではない」ことを第3者に証明することが可能であると、否認していな  
い者が疑いをかけられてしまうことになる。

#### 【0044】

そのため、本発明の別の目的としては、否認データは、リング署名の署名者は  
作成することができないようにすることを目的とする。

#### 【0045】

##### 【課題を解決するための手段】

本発明の目的を達成するために、例えば本発明の連鎖型署名作成方法は以下の  
構成を備える。

#### 【0046】

すなわち、N個の公開鍵とそのうちの1つの公開鍵に対応する秘密鍵によって  
作成可能で、N個の公開鍵のそれぞれについて署名の検証が可能で、N人のメン  
バーのうち誰が署名したかを秘匿することのできる連鎖型署名において、前記連  
鎖型署名データの作成者以外のユーザが署名していないことを検証できるように  
するための否認データを生成する工程を備えることを特徴とする。

#### 【0047】

##### 【発明の実施の形態】

以下添付図面を参照して、本発明を好適な実施形態に従って詳細に説明する。

**【 0 0 4 8 】****[第 1 の実施形態]**

本実施形態に係る連鎖型署名作成処理、連鎖型署名検証処理を実行する装置には例えば図 1 に示す基本構成を備えるコンピュータが適用できる。以下、このコンピュータの基本構成について図 1 を参照して説明する。

**【 0 0 4 9 】**

すなわち、コンピュータ 1 0 0 は図 1 に示すように、公衆回線等のモデム 1 1 8、表示部としてのモニタ 1 0 2、CPU 1 0 3、ROM 1 0 4、RAM 1 0 5、HDD（ハードディスクドライブ）1 0 6、ネットワークへのネットワーク接続部 1 0 7、CD-ROMドライブ 1 0 8、FD（フレキシブルディスク）ドライブ 1 0 9、DVD（デジタル・ビデオ・ディスク、またはDigital Versatile Disk）-ROMドライブ 1 1 0、プリンタ 1 1 5のインターフェース（I/F）1 1 7、及び操作部としてのマウス 1 1 2やキーボード 1 1 3等のインターフェース（I/F）1 1 1が、バス 1 1 6を介して互いに通信可能に接続されて構成されている。

**【 0 0 5 0 】**

マウス 1 1 2 及びキーボード 1 1 3 は、コンピュータ 1 0 0 に対する各種指示等をユーザが入力するための操作部である。この操作部を介して入力された情報（操作情報）は、インターフェース 1 1 1 を介して、CPU 1 0 3 に通知される。

**【 0 0 5 1 】**

コンピュータ 1 0 0 内に保持されている各種情報（文字情報や画像情報等）は、プリンタ 1 1 5 により印刷出力できるようになされている。

**【 0 0 5 2 】**

モニタ 1 0 2 は D R T や液晶画面などにより構成されており、ユーザへの各種指示情報や、文字情報或いは画像情報等の各種情報の表示を行う。

**【 0 0 5 3 】**

CPU 1 0 3 は、コンピュータ 1 0 0 全体の動作制御を司るものであり、以下説明する連鎖型署名作成処理、連鎖型署名検証処理を実行するものである。CP

U103は、HDD106やCD-ROMドライブ108、FDドライブ109、DVD-ROMドライブ110等からRAM105にロードされた各種の処理プログラム（ソフトウェアプログラム）を実行することで、各種の処理を行う。

【0054】

ROM104は、署名の作成・検証処理のための処理プログラム等の各種処理プログラムや、各種データ等を記憶する。

【0055】

RAM105は、CPU103での各種処理のために、一時的に処理プログラムや処理対象の情報を格納するための作業用エリア等を備えるメモリである。

【0056】

HDD106は、大容量記憶装置の一例としての構成要素であり、文字情報や画像情報、或いは各種処理の実行時にRAM105等へ転送される情報変換処理等のための処理プログラム等を保存する。

【0057】

CD-ROMドライブ108は、外部記憶媒体の一例としてのCD-ROM（CD-R）に記憶されたデータを読み込み、また、CD-Rへデータを書き出す機能を有する。

【0058】

FDドライブ109は、外部記憶媒体の一例としてのFD（フレキシブルディスク）に記憶されたデータを読み出す。また、種々のデータをFDへ書き込む機能を有している。

【0059】

DVD-ROMドライブ110は、外部記憶媒体の一例としてのDVDに記憶されたデータを読み出し、また、DVDへデータを書き込む機能を有している。

【0060】

なお、CD、FD、DVD等の外部記憶媒体に対して、例えば、編集用のプログラム或いはプリンタドライバが記憶されている場合には、これらのプログラムをHDD106へインストールしておき、必要に応じて、RAM105へ転送するように構成してもよい。

## 【0061】

インターフェース (I/F) 111は、マウス112或いはキーボード113によるユーザからの入力を受け付けるためのものである。

## 【0062】

モデム118は、通信モデムであり、インターフェース (I/F) 119を介して、例えば、公衆回線等を通じて外部のネットワークに接続される。

## 【0063】

ネットワーク接続部107は、インターフェース (I/F) 114を介して、外部のネットワークに接続される。

## 【0064】

以上の構成を備えるコンピュータによって、本実施形態に係る連鎖型署名作成処理、連鎖型署名検証処理を実行するが、夫々の処理を実行する装置は同一のものであっても良いし、別個の装置であっても良い。

## 【0065】

次に、リング署名に対する否認データ作成処理について説明する。

## 【0066】

[否認データ作成] Schnorrリング署名に対する否認データの作成手順

否認データ作成者は、公開鍵  $y_i$  に対する秘密鍵  $x_i$  を保持するとする。

## 【0067】

1)  $\alpha^* := s_i + x_i \cdot c_i$  とおく。

## 【0068】

2)  $r$  を  $Z_{(q_i)}$  から任意に選択し、 $T^* := g_i^r$  とおく。 $c_i^* := H(M \parallel T^* \parallel T_{(i-1)} \parallel Rep)$  とおく。  
ここで  $Rep$  は否認する趣旨が記述された誓約データである。

## 【0069】

3)  $s_i^* := r - \alpha^* \cdot c_i^* \pmod{q_i}$  とおき、リング署名 ( $c_0, s_0, s_1, \dots, s_{(n-1)}$ ) に対する否認データ ( $s_i^*, c_i^*$ ) を作成する。

## 【0070】

〔否認データ検証〕 Schnorrリング署名に対する否認データの検証手順

否認データ ( $s_i^*$ ,  $c_i^*$ ) に対し、 $T^* := g_i s_i^* (T_{i-1}) c_i^* \bmod p_i$  とおき、 $c_i^* = H(M \parallel T^* \parallel T_{i-1} \parallel Rep)$  が成立するか検証する。

## 【0071】

図2は、リング署名に対する否認データ作成処理を行うための装置の機能構成、もしくはこの処理をコンピュータに実行させるためのプログラムの機能構成を示す図である。本実施形態では同図に示した各部の機能をプログラムにより表現し、このプログラムをコンピュータ100に読み込ませることで、コンピュータ100に同処理を行わせる。

## 【0072】

否認データ作成者は、公開鍵  $y_i$  に対する秘密鍵  $x_i$  をコンピュータ100に接続されたHDD106やCD-ROM、FD、DVD-ROM等に保持させておき、必要に応じてRAM105にロード可能なようにしておく。

## 【0073】

上記〔否認データ作成〕の第1の処理を行うために、リング署名データ  $S$  が入力され、付随データ抽出モジュール204においてリング署名データ  $S$  から  $s_i$ ,  $c_i$  を抽出する。抽出された  $s_i$ ,  $c_i$  及び上記  $x_i$  から、 $\alpha^* := s_i + x_i c_i$  を計算する。

## 【0074】

上記〔否認データ作成〕の第2の処理を行うために、 $r$  を  $Z(q_i)$  から任意に選択し、 $T^* := g_i^r$  を計算し、被署名データ  $M$  が入力され、同時に付随データ抽出モジュール204において  $T_{i-1}$  が抽出され、次に誓約データ添付モジュール203において、被署名データ  $M$  に対して、 $T_{i-1}$  と誓約データ  $Rep$  が付加され、ハッシュ再計算モジュール205に渡され、 $c_i^* := H(M \parallel T^* \parallel T_{i-1} \parallel Rep)$  を計算する。ここで  $Rep$  は否認する趣旨が記述された誓約データである。

## 【0075】

上記「否認データ作成」の第3の処理を行うために、付随データ抽出モジュール204から得た $\alpha^*$ およびハッシュ再計算モジュール205から得た $c\_i^*$ に基づき、再署名モジュール206にて $s\_i^* := r - \alpha^* \cdot c\_i^* \pmod{q\_i}$ を計算し、結果として否認データ $R = (s\_i^*, c\_i^*)$ が出力される。

## 【0076】

図3は否認データを作成する処理のフローチャートである。各ステップにおける処理は上述の通りであるので、ここでの説明は簡単なものにする。また、同図のフローチャートに従ったプログラムはHDD106やCD-ROMドライブ108、FDドライブ109、DVD-ROMドライブ110等からRAM105にロードされ、CPU103がこれを実行することで、コンピュータ100は同図のフローチャートに従った処理、すなわち、否認データを作成する処理を実行することができる。

## 【0077】

付随データ抽出モジュール204はステップS301における付随データの抽出処理を行い、誓約データ添付モジュール203はステップS302における誓約データの添付処理を行い、ハッシュ再計算モジュール205はステップS303におけるハッシュ再計算処理を行い、再署名モジュール206はステップS304における署名の再計算処理を行う。

## 【0078】

リング署名 $(c\_0, s\_0, s\_1, \dots, s\_(n-1))$ の中に含まれている偽造された署名 $s\_i$ を $s\_i^*$ で置き換えることで否認を宣言している。この $s\_i^*$ を生成する操作は、公開鍵 $y\_i$ に対する秘密鍵 $x\_i$ を保持している者にしか作成できない。なぜならば、上記「否認データ作成」における第1の処理は、秘密鍵 $x\_i$ を保持する者しか計算できず、また第3の処理は通常の署名操作と同一であり、秘密データ $\alpha^*$ を保持する者しか計算できないためである。

## 【0079】

また本実施形態では、 $c\_i^*$ の計算においてハッシュ関数に渡すデータとして $T\_ (i-1)$ や $Rep$ を含んでいるが、これらが包含されることは必須ではなく、第1の処理で得られた $\alpha^*$ を用いて再署名を施すことが安全性の根拠となる。したがって $c\_i^*$ の計算には何をハッシュ計算の対象とするかは他にもあらゆるバリエーションが容易に考えられる。

#### 【0080】

##### [第2の実施形態]

第1の実施形態では、作成された否認データをオフラインで検証する方式であったが、本実施形態では、インタラクティブに否認するプロトコルについて示す。

#### 【0081】

[否認するユーザUとそれを検証するユーザVの間のプロトコル]

1)  $V \rightarrow U$ : リング署名 ( $c\_0, s\_0, s\_1, \dots, s\_ (n-1)$ ), チャレンジデータ  $r$  を送付。

#### 【0082】

2)  $U \rightarrow V$ : 以下のように計算された  $s\_i^*$  を送付。リング署名データから  $s\_i, c\_i$  を抽出し、 $\alpha^* := s\_i + x\_i \cdot c\_i$  とおき、 $c\_i^* := H(M \parallel T^* \parallel T\_ (i-1) \parallel r)$  に対し、 $s\_i^* := r - \alpha^* \cdot c\_i^* \bmod q\_i$  を計算する。

#### 【0083】

3)  $V$ :  $c\_i^* := H(M \parallel T^* \parallel T\_ (i-1) \parallel r)$  に対し、 $c\_i^* = H(M \parallel T^* \parallel T\_ (i-1) \parallel Rep)$  が成立するかどうかを検証する。検証OKであれば、ユーザUは、リング署名の作成者ではないことを証明したことになる。

#### 【0084】

図4は、上記プロトコルの処理を表す図である。ステップS401では上記プロトコル1)の処理を、ステップS402、S403では上記プロトコル2)の処理を、ステップS404では上記プロトコル3)の処理を行っている。

#### 【0085】

さらに、上記プロトコルでは  $s_i^*$  が通信上流れるが、ゼロ知識証明プロトコルを用いて対話的に証明する方法でもよい。具体的には、 $\alpha^*$  を計算できるのは秘密鍵  $x_i$  を保持している者だけであるので、 $g^{\alpha^*}$  を公開し、それに対応する  $\alpha^*$  を持っているかどうかを対話的に証明すればよい。

#### 【0086】

##### [第3の実施形態]

上記実施形態においては、Schnorr署名に対するリング署名をベースにしていたが、本実施形態ではDSA署名に対する拡張例を示す。この拡張方法は他の既存の署名方式に対しても適用できる。

#### 【0087】

##### [DSA署名]

Federal Information Processing Standards (FIPS) 186-2, Digital Signature Standard (DSS), January 2000に記載の方式を説明する。記号の表記方法はSchnorr署名での記号に準じる。

#### 【0088】

##### [DSA署名作成] 文書Mに対する署名の作成手順

1)  $\alpha$  を  $Z_q$  から任意に選択し、 $T := (g^\alpha \bmod p) \bmod q$  とおく。

#### 【0089】

2)  $c := H(M)$  とおく。

#### 【0090】

3)  $s := \alpha^{-1} (c + xT) \bmod q$  とおき、 $(s, T)$  を署名データとする。

#### 【0091】

##### [DSA署名検証] 文書Mに対する署名 $(s, T)$ の検証手順

$T = (g^{h(M)} s^{-1} y^{Ts^{-1}} \bmod p) \bmod q$  かどうか検証する。

#### 【0092】

##### [DSAリング署名作成] 文書Mに対する署名の作成手順

1)  $Z_{\_i}$  ( $q_{\_i}$ ) から任意に選択し、 $T_{\_i} := (g_{\_i}^{\alpha} \bmod p_{\_i}) \bmod q_{\_i}$  とおく。

【0093】

2)  $c_{\_i+1} := H(M \parallel T_{\_i})$  とおく。

【0094】

3)  $j = i + 1, \dots, n - 1$  (昇順) について、以下を繰り返す。 $s_{\_j}$  を  $Z_{\_j}$  ( $q_{\_j}$ ) から任意に選択し、 $T_{\_j} := g_{\_j}^{c_{\_j} s_{\_j}^{-1} y_{\_j}} T_{\_j} s_{\_j}^{-1} \bmod p_{\_j}$ ,  $c_{\_j+1} := H(M \parallel T_{\_j})$  とおく。

【0095】

4)  $s_{\_i} := \alpha^{-1} (c_{\_i} + x_{\_i} T_{\_i}) \bmod q$  とおき、 $(c_{\_0}, s_{\_0}, s_{\_1}, \dots, s_{\_n})$  を署名データとする。

【0096】

[DSAリング署名検証] 文書Mに対する署名  $(c_{\_0}, s_{\_0}, s_{\_1}, \dots, s_{\_n})$  の検証手順

1)  $j = 0, \dots, n - 1$  (昇順) について、以下を繰り返す。 $T_{\_j} := g_{\_j}^{c_{\_j} s_{\_j}^{-1} y_{\_j}} T_{\_j} s_{\_j}^{-1} \bmod p_{\_j}$ ,  $c_{\_j+1} := H(M \parallel T_{\_j})$  とおく。

【0097】

2)  $c_{\_n} = c_{\_0}$  かどうかを検証する。

【0098】

上記の方法のほかにも、 $c_{\_i}$  を連鎖させるのではなく、 $T_{\_i}$  を連鎖させる方法でも実現可能である。

【0099】

[第4の実施形態]

上記実施形態では誓約データ Rep を必要としていたが、事前計算データ  $T_{\_j}$  で代用する例を示す。第1の実施形態における「否認データ作成」の第2の操作において、 $c_{\_i}^* := H(M \parallel T_{\_i} \parallel Rep)$  の代わりに、Rep を使用しないで、 $c_{\_i}^* := H(M \parallel T_{\_i})$

などのように  $T\_j$  ( $j \neq i$ ) を用いることも可能である。

#### 【0100】

さらに、同じメッセージに対する連鎖型署名を複数作成し、ハッシュ対象データに先に生成した連鎖型署名データを含めることも可能である。たとえば2つの連鎖型署名を作成する際に、まず  $H(M \parallel T\_i \parallel Rep)$  などと  $Rep$  もハッシュ対象にした第1の連鎖型署名データ ( $c\_0, s\_0, s\_1, \dots, s\_(n-1)$ ) を作成する。次に  $R\_1 := H((c\_0, s\_0, s\_1, \dots, s\_(n-1)))$  とし、 $H(M \parallel T\_i \parallel R\_1)$  などとして第2の連鎖型署名データを作成する。公開時には  $Rep$  を秘匿しておき、 $R\_1$  と第2の連鎖型署名データを公開する。公開後に、否認署名を行いたいエンティティが登場したときにはじめて第1の連鎖型署名データと  $Rep$  を公開し、第1の連鎖型署名データ、第2の連鎖型署名データそれぞれから  $\alpha^*$  を算出して否認署名データの作成にもちいることが可能となる。

#### 【0101】

##### [その他の実施形態]

本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記録媒体（または記憶媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記録媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記録媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記録した記録媒体は本発明を構成することになる。

#### 【0102】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（OS）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0103】

さらに、記録媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0104】

本発明を上記記録媒体に適用する場合、その記録媒体には、先に説明したフローチャートに対応するプログラムコードが格納されることになる。

#### 【0105】

以下に本発明の実施態様の例を示す。

#### 【0106】

〔実施態様1〕 N個の公開鍵とそのうちの1つの公開鍵に対応する秘密鍵によって作成可能で、N個の公開鍵のそれぞれについて署名の検証が可能で、N人のメンバーのうち誰が署名したかを秘匿することのできる連鎖型署名において、前記連鎖型署名データの作成者以外のユーザが署名していないことを検証できるようにするための否認データを生成する手段を備えることを特徴とする連鎖型署名作成装置。

#### 【0107】

〔実施態様2〕 メッセージにデジタル署名を施す際に、事前計算データをメッセージとともにハッシュ関数で圧縮するデジタル署名方式において、N個の公開鍵とそのうち少なくとも一つの秘密鍵を有し、第1の事前計算データを生成し、前記メッセージと第iの事前計算データを含むデータに対して第iのハッシュ値を計算するハッシュ計算手段と、

第iのハッシュ値に署名が施されたように見せかけるように第iの事前計算データと第iの署名データを擬似的に計算する擬似計算手段と、

前記擬似計算手段を逐次的に繰り返して得られた第Nのハッシュ値に対して、第1の事前計算データに呼応した第1の署名データを前記秘密鍵から生成する署名手段と

を備えることを特徴とする連鎖型署名作成装置。

#### 【0 1 0 8】

〔実施態様 3〕 メッセージにデジタル署名を施す際にメッセージだけをハッシュ関数で圧縮してから演算を行うデジタル署名方式において、

該デジタル署名方式を、事前計算データを計算してメッセージとともにハッシュ関数で圧縮するデジタル署名方式に変更して実行することを特徴とする実施態様 2 に記載の連鎖型署名作成装置。

#### 【0 1 0 9】

〔実施態様 4〕 実施態様 2 又は 3 に記載の連鎖型署名作成装置で生成された連鎖型署名データに対して、前記連鎖型署名データの作成者以外のユーザが署名していないことを検証できるようにするために、否認データを生成する手段を有することを特徴とする連鎖型署名作成装置。

#### 【0 1 1 0】

〔実施態様 5〕 更に、

署名対象メッセージを受領する署名対象メッセージ受領手段と、

前記署名対象メッセージに対する連鎖型署名が施された連鎖型署名データを受領する連鎖型署名データ受領手段と、

前記署名対象メッセージに対して誓約データを付加する誓約データ付加手段と、

前記連鎖型署名データから署名の再計算に必要なデータを抽出する付随データ抽出手段と、

前記誓約データ付加手段で作成された誓約データ付加済みメッセージに対して改めて署名を施す再署名手段と、

前記再署名手段で計算されたデータを出力する否認データ出力手段とを備えることを特徴とする実施態様 4 に記載の連鎖型署名作成装置。

#### 【0 1 1 1】

〔実施態様 6〕 前記再署名手段は、

前記誓約データ付加手段で得られたデータのハッシュ値を改めて計算するハッシュ再計算手段と、

前記ハッシュ再計算手段で計算されたハッシュ値に対して演算を施す演算手段と

を備えることを特徴とする実施態様 5 に記載の連鎖型署名作成装置。

#### 【0112】

〔実施態様 7〕 前記誓約データは事前計算データで代用することを特徴とする実施態様 5 に記載の連鎖型署名作成装置。

#### 【0113】

〔実施態様 8〕 前記第 1 の事前計算データは位数  $P-1$  の乗法群（ただし  $P$  は素数）の生成元  $g$  に対して、擬似乱数  $k$  ( $< P-1$ ) を生成し、 $g^k \pmod{P}$  の演算を行い、この演算結果を第 1 の事前計算データとすることを特徴とする実施態様 2 乃至 7 の何れか 1 項に記載の連鎖型署名作成装置。

#### 【0114】

〔実施態様 9〕 前記デジタル署名方式は離散対数問題に安全性を置くデジタル署名方式であることを特徴とする実施態様 1 乃至 8 の何れか 1 項に記載の連鎖型署名作成装置。

#### 【0115】

〔実施態様 10〕 前記否認データは対話的なやり取りで証明することを特徴とする実施態様 1 乃至 9 の何れか 1 項に記載の連鎖型署名作成装置。

#### 【0116】

〔実施態様 11〕 メッセージにデジタル署名を施す際に事前計算データをメッセージとともにハッシュ関数で圧縮するデジタル署名方式において、 $N$  個の公開鍵を有し、前記メッセージと第  $i$  の事前計算データを含むデータに対して第  $i$  のハッシュ値を計算するハッシュ計算手段と、

第  $i$  の署名データに検証のための演算を施す検証演算手段と、

前記検証演算手段を逐次的に繰り返して得られた第  $N$  のハッシュ値が第 1 のハッシュ値と一致するかどうかを検証する検証手段と

を備えることを特徴とする連鎖型署名検証装置。

#### 【0117】

〔実施態様 12〕 メッセージにデジタル署名を施す際にメッセージだけを

ハッシュ関数で圧縮してから演算を行うデジタル署名方式において、該デジタル署名方式を、事前計算データを計算してメッセージとともにハッシュ関数で圧縮するデジタル署名方式に変更して実行することを特徴とする実施態様 11 に記載の連鎖型署名検証装置。

#### 【0118】

〔実施態様 13〕 実施態様 1 乃至 3 の何れか 1 項に記載の連鎖型署名作成装置で生成された連鎖型署名データに対して、前記連鎖型署名データの作成者以外のユーザが署名していないことを検証できるようにするために否認データを生成する手段を更に備えることを特徴とする実施態様 11 又は 12 に記載の連鎖型署名検証装置。

#### 【0119】

〔実施態様 14〕 更に、  
署名対象メッセージを受領する署名対象メッセージ受領手段と、  
前記署名対象メッセージに対する連鎖型署名が施された連鎖型署名データを受領する連鎖型署名データ受領手段と、  
前記連鎖型署名データ受領手段に対する否認データを受領する否認データ受領手段と、  
前記否認データに対応する誓約データを受領する誓約データ受領手段と、  
前記連鎖型署名データから検証に必要なデータを抽出する付随データ抽出手段と、  
前記署名対象メッセージ、上記誓約データからハッシュ値を計算するハッシュ演算手段と、  
前記否認データに前記公開鍵で演算を施し、前記ハッシュ演算手段で得られたデータと一致するかどうかを検証する否認データ検証手段と  
を備えることを特徴とする実施態様 13 に記載の連鎖型署名検証装置。

#### 【0120】

〔実施態様 15〕 前記デジタル署名方式は離散対数問題に安全性を置くデジタル署名方式であることを特徴とする実施態様 11 乃至 14 の何れか 1 項に記載の連鎖型署名検証装置。

## 【0121】

〔実施態様16〕 前記否認データは対話的なやり取りで証明することを特徴とする実施態様11乃至15の何れか1項に記載の連鎖型署名検証装置。

## 【0122】

〔実施態様17〕 実施態様1乃至8の何れかに記載の連鎖型署名作成装置と、実施態様11乃至16の何れかに記載の連鎖型署名検証装置とを有することを特徴とする連鎖型署名システム。

## 【0123】

〔実施態様18〕 N個の公開鍵とそのうちの1つの公開鍵に対応する秘密鍵によって作成可能で、N個の公開鍵のそれぞれについて署名の検証が可能で、N人のメンバーのうち誰が署名したかを秘匿することのできる連鎖型署名において、前記連鎖型署名データの作成者以外のユーザが署名していないことを検証できるようにするための否認データを生成する工程を備えることを特徴とする連鎖型署名作成方法。

## 【0124】

〔実施態様19〕 メッセージにデジタル署名を施す際に、事前計算データをメッセージとともにハッシュ関数で圧縮するデジタル署名方式において、N個の公開鍵とそのうち少なくとも一つの秘密鍵を有し、第1の事前計算データを生成し、前記メッセージと第iの事前計算データを含むデータに対して第iのハッシュ値を計算するハッシュ計算工程と、

第iのハッシュ値に署名が施されたように見せかけるように第iの事前計算データと第iの署名データを擬似的に計算する擬似計算工程と、

前記擬似計算工程における処理を逐次的に繰り返して得られた第Nのハッシュ値に対して、第1の事前計算データに呼応した第1の署名データを前記秘密鍵から生成する署名工程と

を備えることを特徴とする連鎖型署名作成方法。

## 【0125】

〔実施態様20〕 メッセージにデジタル署名を施す際に事前計算データをメッセージとともにハッシュ関数で圧縮するデジタル署名方式において、N個の

公開鍵を有し、前記メッセージと第  $i$  の事前計算データを含むデータに対して第  $i$  のハッシュ値を計算するハッシュ計算工程と、  
第  $i$  の署名データに検証のための演算を施す検証演算工程と、  
前記検証演算工程における処理を逐次的に繰り返して得られた第  $N$  のハッシュ値が第 1 のハッシュ値と一致するかどうかを検証する検証工程と  
を備えることを特徴とする連鎖型署名検証方法。

#### 【0 1 2 6】

〔実施態様 2 1〕 コンピュータを実施態様 1 乃至 1 0 の何れか 1 項に記載の連鎖型署名作成装置として機能させることを特徴とするプログラム。

#### 【0 1 2 7】

〔実施態様 2 2〕 コンピュータを実施態様 1 1 乃至 1 6 の何れか 1 項に記載の連鎖型署名検証装置として機能させることを特徴とするプログラム。

#### 【0 1 2 8】

〔実施態様 2 3〕 コンピュータに実施態様 1 8 又は 1 9 に記載の連鎖型署名作成を実行させることを特徴とするプログラム。

#### 【0 1 2 9】

〔実施態様 2 4〕 コンピュータに実施態様 2 0 に記載の連鎖型署名検証方法を実行させることを特徴とするプログラム。

#### 【0 1 3 0】

〔実施態様 2 5〕 実施態様 2 1 乃至 2 4 の何れか 1 項に記載のプログラムを格納することを特徴とするコンピュータ読み取り可能な記憶媒体。

#### 【0 1 3 1】

#### 【発明の効果】

以上説明したように、本発明によれば、勝手に流用された公開鍵に対する秘密鍵を保持するユーザ自らが署名していないことを示す（否認する）ことができない問題を解決し、ユーザが署名を作成していないことを示す否認データを作成することができる。

#### 【図面の簡単な説明】

#### 【図 1】

本発明の実施形態に係る連鎖型署名作成処理、連鎖型署名検証処理を実行する装置に適用するコンピュータの基本構成を示す図である。

【図 2】

リング署名に対する否認データ作成処理を行うための装置の機能構成、もしくはこの処理をコンピュータに実行させるためのプログラムの機能構成を示す図である。

【図 3】

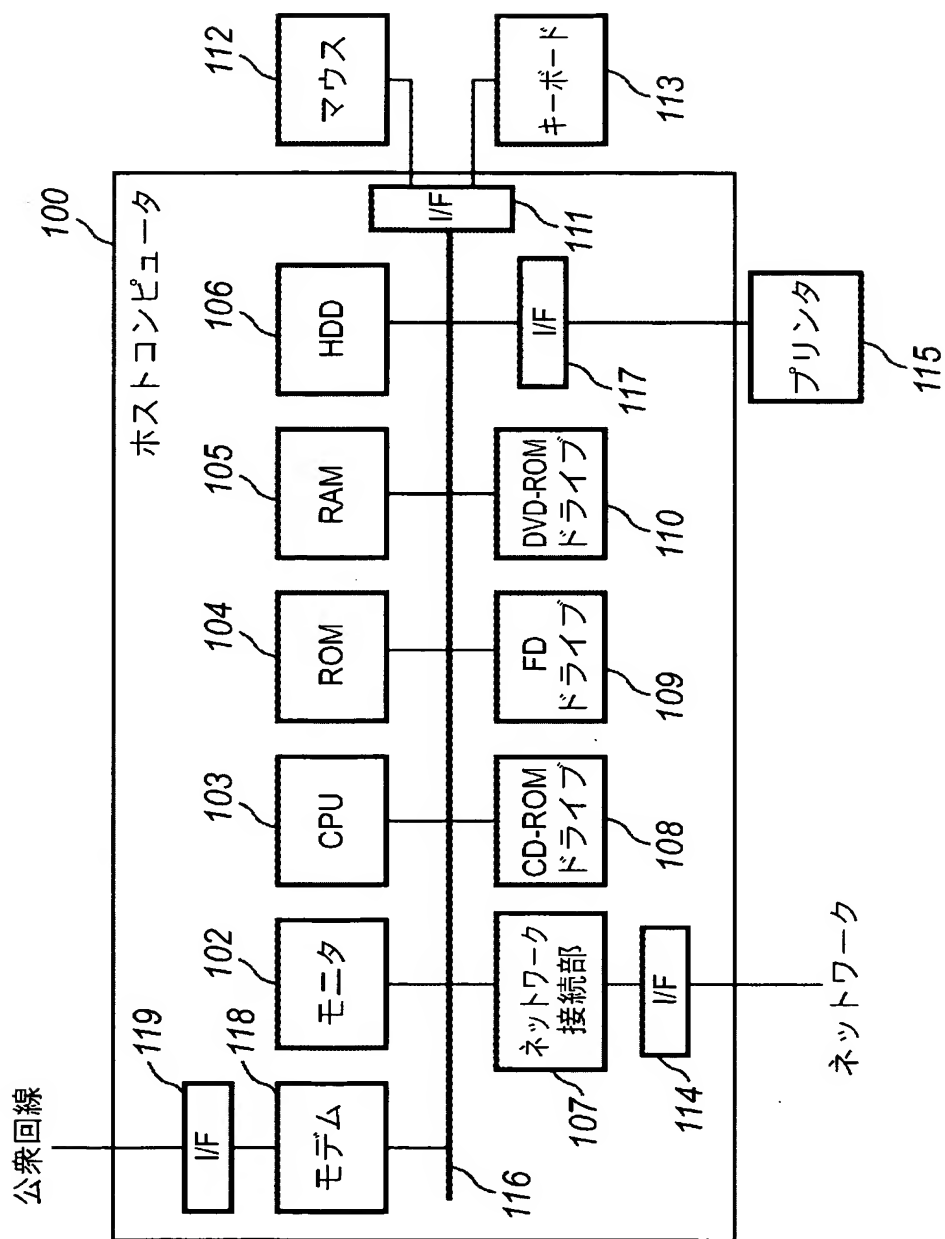
否認データを作成する処理のフローチャートである。

【図 4】

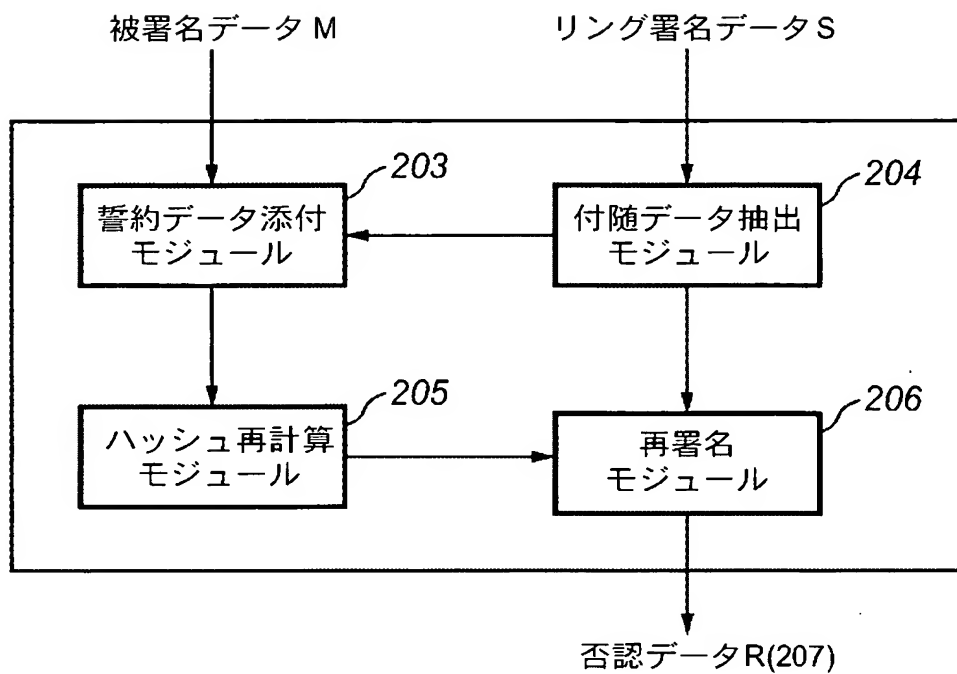
インタラクティブに否認するプロトコル処理を表す図である。

【書類名】 図面

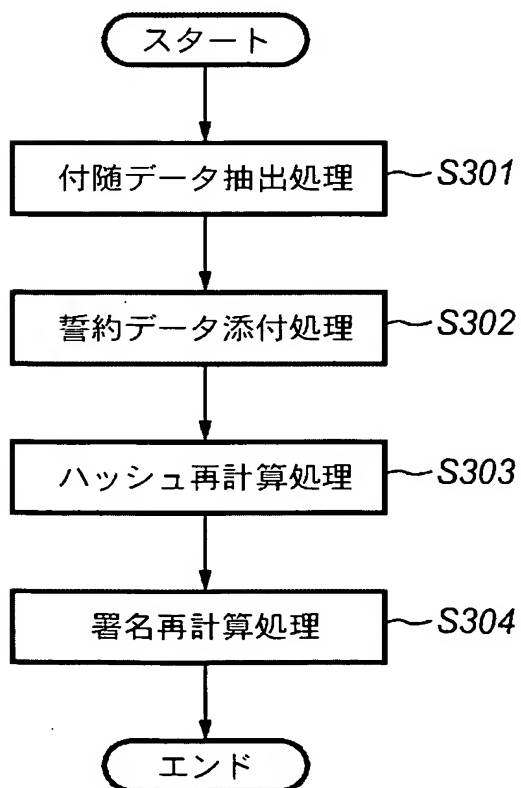
【図 1】



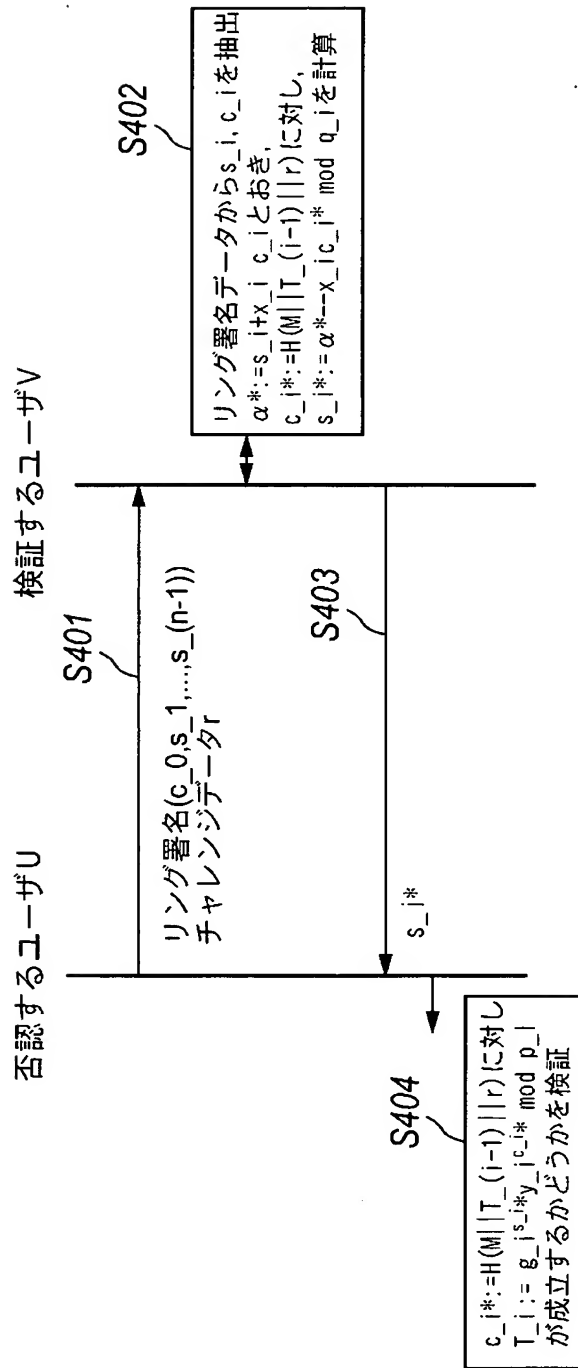
【図 2】



【図 3】



【図 4】



【書類名】 要約書

【要約】

【課題】 連鎖型署名において、勝手に流用された公開鍵に対する秘密鍵を保持するユーザ自らが署名していないことを示す（否認する）ことができない問題を解決し、ユーザが署名を作成していないことを示す否認データを作成すること。

【解決手段】 モジュール 2 0 4 はリング署名データ S から  $s_i, c_i$  を抽出し、 $s_i, c_i$ 、秘密鍵  $x_i$  から  $\alpha^* := s_i + x_i c_i$  を計算する。被署名データ M が入力され、同時にモジュール 2 0 4 において  $T_{(i-1)}$  が抽出され、モジュール 2 0 3 は M に  $T_{(i-1)}$  と誓約データ Rep を付加し、モジュール 2 0 5 は  $c_i^* := H(M || T_{(i-1)} || Rep)$  を計算する。 $\alpha^*, c_i^*$  およびランダム値  $r$  に基づき、モジュール 2 0 6 は  $s_i^* := r - \alpha^* c_i^* \bmod q_i$  を計算し、否認データ R を出力する。

【選択図】 図 2

特願 2 0 0 3 - 0 1 6 7 1 8

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 1 0 0 7 ]

1. 変更年月日	1 9 9 0 年 8 月 3 0 日
[変更理由]	新規登録
住 所	東京都大田区下丸子 3 丁目 3 0 番 2 号
氏 名	キャノン株式会社